

# Examen de Conception de Circuits

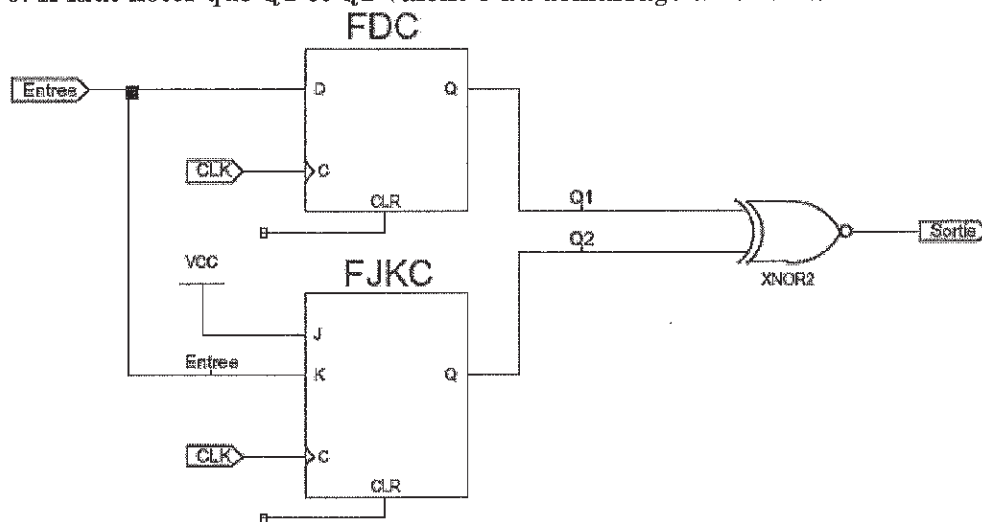
E. Mesnard  
18 mai 2010

Documents de cours et calculatrices **autorisés**.  
Durée : **2 heures**

## Exercice 1 (4 points)

### Analyse d'un circuit

Tracer le chronogramme des signaux  $Q1$ ,  $Q2$  et *Sortie*, sur 10 cycles d'horloge (CLK) sachant que l'entrée *Entrée* vaut 0 pendant 3 cycles, puis, passe à 1 pendant 3 autres cycles avant de revenir à 0. Il faut noter que  $Q1$  et  $Q2$  valent 0 au démarrage du circuit.



## Exercice 2 (7 points) Synthèse micro-programmée à aiguillages

Soit l'automate simplifié d'un processeur, dont les logiques de transitions et de sorties sont données ci-après. Outre les passages en séquence et les sauts conditionnels et inconditionnels, cet automate contient deux **aiguillages à 4 branches** (nommés *Aigui. 1* et *Aigui. 2* et chargés d'aiguiller en fonction des variables *Longueur(1:0)* et *Traitement(1:0)*).

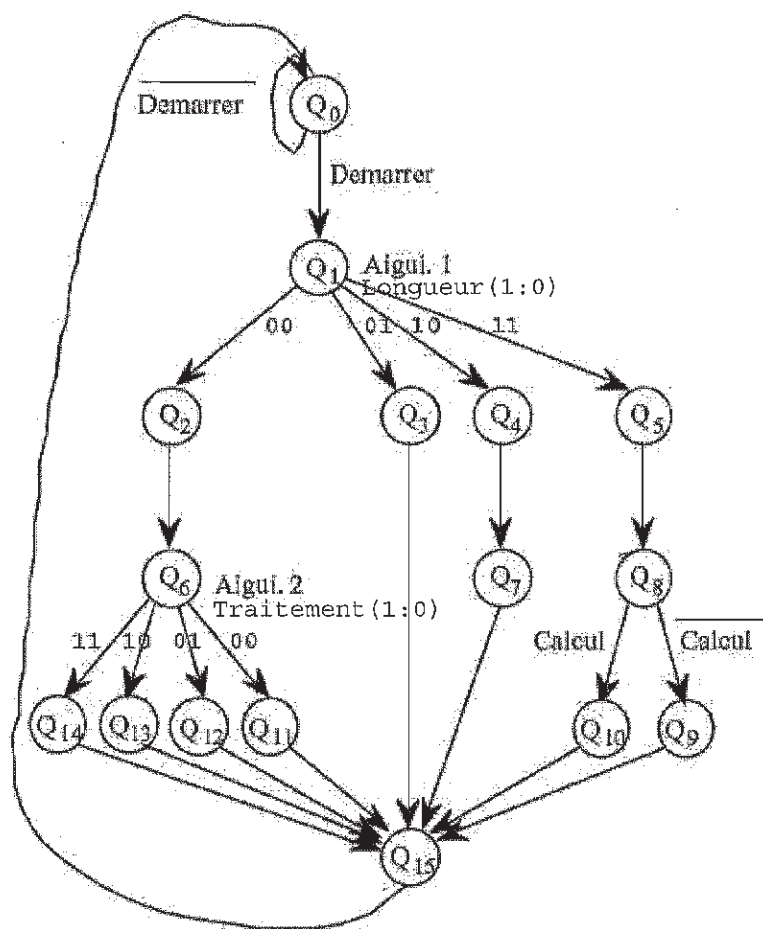
L'objectif ici est de tirer partie de cette singularité et **modifier** alors la technique de synthèse  **$\mu$ -programmée** pour traiter ces aiguillages **différemment** des éclatements classiques.

En effet, les branches de ces aiguillages sont toujours telles que :

- elles sont au nombre de 4,
- elles sont implémentées en mémoire à des adresses successives respectant l'ordre binaire,
- la première adresse de l'aiguillage est donnée dans la  $\mu$ -instruction par le champ de  $\mu$ -adresse « A ».

Se baser sur cette nouvelle instruction **AIG4** (au lieu de ECLT) pour réaliser la synthèse  $\mu$ -programmée modifiée de cet automate. Donner le schéma complet en précisant le contenu de la ROM, ainsi que le  $\mu$ -programme correspondant à cet automate.

### Logique de transition $\tau$



### Logique de sortie $\sigma$

- Q<sub>0</sub> : S<sub>0</sub>
- Q<sub>1</sub> : rien
- Q<sub>2</sub> : S<sub>2</sub>
- Q<sub>3</sub> : S<sub>1</sub> // S<sub>3</sub>
- Q<sub>4</sub> : S<sub>3</sub>
- Q<sub>5</sub> : S<sub>2</sub> // S<sub>3</sub>
- Q<sub>6</sub> : S<sub>2</sub>
- Q<sub>7</sub> : S<sub>3</sub>
- Q<sub>8</sub> : S<sub>1</sub>
- Q<sub>9</sub> : S<sub>1</sub>
- Q<sub>10</sub> : S<sub>1</sub> // S<sub>2</sub>
- Q<sub>11</sub> : S<sub>1</sub> // S<sub>3</sub>
- Q<sub>12</sub> : S<sub>3</sub>
- Q<sub>13</sub> : S<sub>2</sub> // S<sub>3</sub>
- Q<sub>14</sub> : S<sub>2</sub>
- Q<sub>15</sub> : S<sub>1</sub>

### Problème (9 points)

### Détecteur Simple clic / Double clic

Concevoir (intégralement...) un circuit capable de déterminer le type de clic (simple ou double) effectué sur un bouton d'entrée (*Bouton*, sur BTN0). Le résultat est lisible par la présence d'un « 1 » soit sur la sortie *Simple\_Clic*, soit sur la sortie *Double\_Clic*. Ces sorties présentent les valeurs 0 lorsque le détecteur est en phase d'attente ou d'analyse. Une entrée d'acquiescement (*Ack\_Clic*, sur BTN1) permet de remettre le détecteur en position initiale d'attente de clic(s). Une horloge, de période 20ms, ainsi que le jeton associé sont disponibles. Considérer qu'un double clic correspond à deux appuis successifs sur le bouton dans un intervalle de temps **inférieur à 640ms**.

Dans le cas contraire, l'analyse doit conduire à un simple clic (soit, car il n'y a eu effectivement qu'un seul appui sur le bouton, soit parce que le second appui a été fait au-delà de 640ms après le premier appui).

